REMARKS

Upon entry of this amendment which amends claims 2, 6, 10, 31, 35, 50, and 61, and adds new claim 62, claims 2-4, 6-19, 21-33, and 35-62 will be pending.

In the office action, claims 2-4, 6-19, 21-28, 30-32, and 35-44 were rejected under 35 U.S.C. §103(a) as being unpatentable over Nachenberg (U.S. Patent No. 5,826,013, hereinafter Nachenberg '013) in view of Nachenberg (U.S. Patent No. 6,021,510, hereinafter Nachenberg '510) and Das et al. (U.S. Patent No. 7,367,057, hereinafter "Das"). Claims 29, 33, 46-47, and 59 were rejected under 35 U.S.C. §103(a) as being unpatentable over Nachenberg '013, Nachenberg '510, Das, and further in view of SimOS. Claims 48-58 and 60-61 were not included under the heading of the rejection, but were addressed under the substantive rejection. A rejection for claim 53 was not found and is respectfully requested. Applicants respectfully request reconsideration of the claims in view of the amendments above and remarks below.

Amendment

Amendments to the claims are made to address concerns raised by the rejection with regard to the Nachenberg '013 and Das references. Although Applicants generally disagree with the rejection's interpretation of the Nachenberg '013 and Das references, an effort is made herein to clarify the claimed invention to bring prosecution to a quick resolution.

The amendments are made to clarify the dynamic verification, as claimed. For example, in claim 2, the dynamic verification is based on a next instruction to be executed in the series of computable-executable instructions in the program. Execution of the program is continued after the verifying is dynamically performed by executing the next instruction as long as the verifying determines that the program is valid. In contrast, Nachenberg '013 does not continue execution of the program after dynamically performing the verifying by executing the instruction. Rather, in Nachenberg '013, the process is ended or the emulation process starts again with a new instruction.

Also, the claimed "ensuring that the program is not executed without dynamically performing the verifying" is clarified to distinguish over Das.

New claim 62 further defines claim 2 and is described in more detail below.

No new matter has been introduced by this amendment.

Claim Rejections - 35 U.S.C. §103(a)

Claims 2-4, 15-19, 21-33, 48-49, and 62

Claim 2 was rejected under 35 U.S.C. §103(a) as being unpatentable over Nachenberg '013, Nachenberg '510, and Das. Applicants submit that Nachenberg '013, Nachenberg '510, and Das do not disclose or suggest "determining an identifying value for a memory block that contains the next instruction." The rejection states that Nachenberg '013 discloses this feature at Fig. 4b, 468, 3 bytes match? - 474, viral signature match. The rejection cites the next instruction being identified in Nachenberg '013 at Fig. 4a, 424, fetch instruction.

Claim 2 recites the next instruction is an instruction that is to be executed after dynamically performing the verifying as long as the verifying determines that the program is valid. Nachenberg '013 discloses an emulation and scanning phase. During the emulation phase, Nachenberg '013 fetches an instruction. See, e.g., Nachenberg '013, col. 11, lines 23-25. The fetched instruction is then emulated and any pages in virtual memory affected by the instruction are tagged. See, e.g., Nachenberg '013, col. 11, lines 60-63. Nachenberg '013 then enters a scanning phase after emulation of sufficient instructions. See, e.g., Nachenberg '013, col. 12, lines 11-19. During the scanning phase, Nachenberg '013 scans the tagged pages. See, e.g., Nachenberg '013, col. 12, lines 31-43. After the scan is completed, Nachenberg '013 returns to the emulation process or may not perform any further emulation. See, e.g., Nachenberg '013, col. 12, lines 53-62. First, Nachenberg '013 may not return to the emulation phase. Second, if Nachenberg '013 returns to the emulation phase, the fetched instruction as cited in the rejection would not be emulated again. Thus, even if emulation is considered execution of a program, Nachenberg '013 does not disclose or suggest identifying value for a memory block that contains the next instruction, where the next instruction is executed upon continuing execution of the program as long as the verifying determines that the program is valid.

Applicants also submit that Nachenberg '013, Nachenberg '510, and Das do not disclose or suggest "ensuring that the program is not executed without dynamically performing the verifying, the verifying based on a next instruction to be executed in the series of computer-executable instructions in the program" and "continuing execution of the program after

dynamically performing verifying by executing the next instruction as long as the verifying determines that the program is valid." (emphasis added). The rejection states that Nachenberg '013 and Nachenberg '510 do not disclose these features, but Das discloses these features at col. 6, line 49 - col.7, line 2. Das discloses an instruction is fetched and transmitted to a virus detection unit, where the instruction is processed to determine whether it is associated with a virus. See Das, e.g., col. 6, lines 51-59. If the instruction is not associated with a virus, the instruction is sent to a fetch and decode unit, and then to the instruction pool. Later, the instruction is selected and executed. See, e.g. Das, col. 7, lines 3-15. Das discloses these steps are described sequentially for one instruction. See, e.g., Das, col. 7, lines 29-30. During the sequential discussion, Das does not disclose or suggest ensuring that the program is not executed. Das just discloses a step by step process for processing an instruction over time without mentioning ensuring the program is not executed.

Also, Das is discussing a conceptual processing of one instruction in col. 6, line 49 - col.7, line 2 and has left out details of how the processor works in this section. A prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away from the claimed invention. *W.L. Gore & Associates, Inc. v. Garlock, Inc.*, 721 F.2d 1540, 220 USPQ 303 (Fed. Cir. 1983), *cert. denied*, 469 U.S. 851 (1984); See also MPEP 2141.02. Das discloses that another instruction is executed contemporaneously by a processor while the virus detection unit is determining whether an instruction is associated with a virus. See, e.g., Das, col. 7, lines 34-40. Thus, not only does Das not disclose or suggest ensuring that the program is not executed without dynamically performing the verifying in the sequential description of processing for one instruction, Das teaches away from ensuring that the program is not executed by teaching a processor that contemporaneously executes an instruction while virus checking.

Accordingly, Applicants respectfully request withdrawal of the rejection of claim 2. Claims 3-4, 15-19, 21-33, 48-49, and 62 depend from claim 2 and thus derive patentability at least therefrom. These claims also recite additional non-obvious and novel features. For example, claim 62 recites "wherein ensuring that the program is not executed without dynamically performing the verifying comprises interrupting execution of the series of computer-executable instructions of the program while dynamically performing the verifying." (emphasis

added). As discussed above, Nachenberg '013 does not disclose or suggest verifying a next instruction as claimed, and Das discloses contemporaneously executing another instruction while performing the virus checking. Further, Das does not disclose or suggest interrupting execution of the program while dynamically performing the verifying. Rather, Das discloses generating an interrupt after the virus checking is performed and determines that a virus is present. See, e.g., Das, Fig. 6, block 607 and 620.

Claims 6-7

Claim 6 recites "executing the current instruction when the verifying determines that the program is valid." As discussed above, Nachenberg '013 does not disclose or suggest executing the current instruction after the verifying. Accordingly, Applicants respectfully request withdrawal of the rejection of claim 6.

Claim 7 depends from claim 6 and thus derives patentability at least therefrom. Claim 7 also recites additional nonobvious and novel features. Claim 7 recites "identifying an indeterminate portion of the current memory block, the indeterminate portion being non-indicative of validity of the current memory block as a whole" and "configuring the mask so that the mask designates at least the indeterminate portion to be ignored when generating the hash value." Applicants have previously argued in the amendment filed October 6, 2009 that Nachenberg '013 does not disclose the above features. Specifically, the excluding of viruses that cannot perform memory writes by Nachenberg '013 does not refer to excluding locations in memory from scanning, but excluding viruses from the database that the tagged memory locations can potentially match. Virus scanning in Nachenberg '013 involves comparing scanned code with known virus signatures to exclude viruses as being possible matches. However, the rejection has stated the Applicant's arguments attack the references individually as the only response. However, in the rejection, on page 7, Nachenberg '013 is only cited as disclosing the elements of claim 7. Further, Applicants submit Nachenberg '510 does not disclose identifying an indeterminate portion of the current memory block and configuring the mask as claimed. Nachenberg '510 is directed to scanning a file and does not disclose configuring a mask for a memory block. Das also does not disclose a mask. Thus, Nachenberg '013, Nachenberg '510, and Das do not disclose or suggest every element of claim 7.

Claims 35-44, 46-47, and 50-61

Applicants submit claims 35-44, 46-47, and 50-61 should be allowable for at least a similar rationale as discussed with respect to claim 2. For example, claim 35 is a machine-readable storage medium claim that includes similar limitations to claim 2. Claims 36-44 and 46-47 depend from claim 35 and thus derive patentability at least therefrom.

Also, claim 50 discloses "if the identifying value satisfies a validation condition, allowing execution after satisfying the validation condition, under control of the system software, of instructions stored in the unvalidated memory block." (emphasis added). Nachenberg '013 does not disclose or suggest identifying the invalidated memory block and then allowing execution after satisfying the validation condition. Accordingly, Applicants respectfully request withdrawal of the rejection of claim 50. Claims 51-61 depend from claim 50 and thus derive patentability at least therefrom.

Conclusion

Since the cited prior art fails to teach or suggest each of the features set forth in the claims for at least the above-listed reasons, Applicants respectfully submit that independent claims 2, 6, 35 and 50 are allowable and hence a Notice of Allowance is earnestly and respectfully requested. Since each of the pending dependent claims include all the features of one of the independent claims, Applicants respectfully submit that the dependent claims are allowable for at least the reasons cited above with regard to the independent claims and further, because they further define and distinguish the invention from the prior art. Also, the above discussion of dependent claims is not exhaustive and merely presents examples of differences between Applicants' claimed invention and cited references. Applicants respectfully request independent reconsideration of the outstanding rejections of the independent claims and each of the dependent claims.

Applicants request reconsideration of the outstanding rejections and issuance of a Notice of Allowance. The Examiner is invited to contact the undersigned at 650-427-2390 to discuss any additional changes the Examiner may feel is necessary in light of this Amendment. If any other fees are due in connection with filing this amendment, the Commissioner is also authorized to charge Deposit Account No. 50-2652 (Order No. A043).

Date: June 2, 2010

Respectfully submitted,

/Brian N. Young/
Brian N. Young
Reg. No. 48,602